RADIO FREQUENCY IDENTIFICATION (RFID) TAGGING SYSTEM FOR NEWBORN BABIES

SYAHIRA NADIA BINTI SHAHARUDDIN

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

RADIO FREQUENCY IDENTIFICATION (RFID) TAGGING SYSTEM FOR NEWBORN BABIES

By

SYAHIRA NADIA SHAHARUDDIN

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in partial fulfillment of the requirement
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

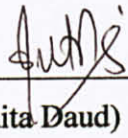
RADIO FREQUENCY IDENTIFICATION (RFID)
TAGGING SYSTEM FOR NEWBORN BABIES

By

Syahira Nadia Shaharuddin

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved by,



(Pn. Hanita Daud)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

(SYAHIRA NADIA SHAHARUDDIN)

ABSTRACT

With recent advances in wireless technology, Radio Frequency Identification (RFID) systems are widely used in a variety of tracking, security and tagging applications. The applications of RFID systems are wide such using in medical history storage, dental prosthesis tracking, toll ways services and animal tracking applications. The project proposes an implementation of RFID Tagging System for Newborn Babies in improving the security features for hospital use. This project will use an active RFID tags since require to detect in a wide range. The tag will be attached to the mother's hand and her newborn baby's ankle right after the delivery the baby in the maternity ward.

ACKNOWLEDGEMENT

Firstly, I would like to thank the Almighty God for giving me the strength and time to complete this project. My greatest gratitude and appreciation is extended to my supervisor, Mrs. Hanita Daud for giving me support, motivation and guidance during the completion of the project. I also would like to express my special million of thanks to beloved my parents and siblings who are always believing in me and continuously supporting all the way. I also would like to give a thousand thanks to the student postgraduate, Elisha for his cooperation, knowledge sharing, endless guidance and advice on the work in order to complete the project successfully. Last but not least, thank you to all lecturers and colleagues who had directly and indirectly contributed to this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic Acid
RFID	Radio Frequency Identification
GUI	Graphical User Interface
VB	Visual Basic
GSM	Global System foe Modem
SMS	Short Messaging Services
HIV	Human immunodeficiency virus
UHF	Ultra High Frequency

CHAPTER 1
INTRODUCTION

1.1 Background of Study

The RFID Tagging System for Newborn Babies is designed for hospital use to avoid babies and mothers from mix-up and to make sure all the newborn babies at the hospital are in a safe condition. The system will monitor, match the babies with the respective mothers and tamper proof strap is used to avoid any attempt to remove the tag. Bracelet will be attached to the mother's hand and her newborn baby's ankle right after delivering the baby in the maternity room. Baby's tag will be monitored by the reader and if the baby's tag has been tampered or baby is out of the range, the RFID will detect it and alert window will be activated.

1.2 Problem Statement

The arrival of a new baby is usually a happy event but it can also be a stressful time to the parents after knowing that they going home with the wrong baby. Exactly how would you feel, knowing this happened to you? Newborn identification is still unsolved problem since methods used nowadays have shown to be inefficient and unsecure. Most of the existing systems in hospitals use paper straps attached at the mother's wrist and her baby's ankle with mother's name handwritten on it and the strap is easy to be removed. Scenario of mother-baby mix-ups in hospitals is more common and has more costly consequences than one might expect. For example it is costly to have a test on a switched newborn baby's Deoxyribonucleic Acid (DNA) to confirm which baby belongs to which mother. Some of the parents may sue the hospital, claiming it was negligence and due to the mother's traumatized by the

incident. Since this scenario will affect the hospital's reputation, hospital is responsible to control, tracking and monitoring of newborn babies that essential for hospital operations. Handling newborn is the responsibility of the hospital until handing over to parents.

Happened at Winchester Hospital, Massachusetts a newborn was given to the wrong mother for breast-feeding. The employee responsible for the mix-up had been fired due to a mistake and the hospital administrators said it was horrible and unacceptable situation. The wrong mother has been tested for HIV because the newborn baby has been exposed to her milk. The testing was done as a provision, and luckily the wrong mother does not have HIV or other infectious diseases.[8] By having a Radio Frequency Identification (RFID) Tagging System for Newborn Babies, this situation can be avoided by tagging the active tag on the mother's hand and her newborn baby's ankle. Mostly bracelet identification is using a paper strap that made from a normal paper and easily to get rid. There will be no other way to trace the mother and the baby if this problem occurred. The idea of this project is using a material that can avoid someone to tamper the strap identification and if the strap has been cut, the system will trigger an alarm to alert the hospital security to this situation.

1.3 Objectives

The objective of RFID Tagging for Mothers and Newborn Babies:

1. To match RFID tag between mother and her newborn baby.
2. To alert the hospital security system if the baby's tag is out of zone. If this scenario happened, the RFID reader will detect it and alert window will be activated.
3. To alert the hospital security system if baby's tag is tampered. The system will activate the alert window when this situation occurs.

1.4 Scope of Study

Basic understanding of the hardware of RFID tag and the reader is very important in this project. Gathering the information and theoretical studies may increase the understanding about the project. Therefore, developing software as an interface between the hardware and the database in a computer is the step that is required in completing this project. Other than that, design RFID-based system for mother's tag and newborn baby's tag using RFID technology and to build a suitable Graphical User Interface (GUI) using Visual Basic and database system using Xampp software are the scope of study of the project. MySQL connector software is to intergrate the interface using Visual Basic with the database from Xampp-win32 software.

CHAPTER 2
LITERATURE REVIEW

2.1 Radio Frequency Identification (RFID).

Radio Frequency Identification (RFID) is a method for remotely storing and retrieving data using devices called RFID tags or transponders. An RFID tag is a small object, such as an adhesive sticker, that can be attached to or incorporated into a product. RFID tags are composed of an antenna connected to an electronic chip. These chips transform the energy of radio-frequency queries from an RFID reader or transceiver to respond by sending back information they enclose.[4] Finally, a computer hosting a specific RFID application or middleware pilots the reader and processes the data it sends. RFID has great characteristics:

1. It may scan tags in motion.
2. Since radio waves can pass through most solid objects, the tags don't need to be in direct line of sight of the RFID reader. Having labeled or tagged objects being identifiable in a ubiquitous and flexible manner is already a good start. Building a network out of these objects, so that with a unique number one can easily retrieve information about them, would enable much more interesting use cases. [4]

A RFID technology is composed by several elements:

1. Readers
2. Tags
3. Software, and security programs for the readers.[9]

2.1.1 RFID Reader.

RFID readers are devices that perform the interrogation of RFID transponders. A RFID system cannot be imagined without the presence of a RFID reader to perform the interrogation and in some cases the power supply of RFID tags. Therefore it could be said that the reader and transponders are in a master-slave relationship where the reader acts as a master and the transponders are slaves. RFID readers themselves are in a slave position as well. A software application that performs the reading of data from the RFID reader acts as the master unit and sends commands to the reader. In a hierarchical system structure the application software represents the master while the reader is a slave.[3]

Antennas also come in a diverse range of form and technical factors. The size could vary from under a square centimeter to several square meters. Technically speaking, UHF reader antennas can be classified as circular-polarized or linear-polarized antenna. The former emit and receive radio waves from all directions, while the later work best in one particular direction. Therefore circular-polarized antennas are less sensitive to transmit readers could come in four types: handheld, vehicle-mount, post-mount, and hybrid.[10]

Instead of visible light used in ordinary bar code labels, these tags use radio waves to communicate with the readers. The readers generate signs that are able, by one hand to supply energy to the tag in order to generate data and, on the other hand, to send a sign of interrogation.[9]

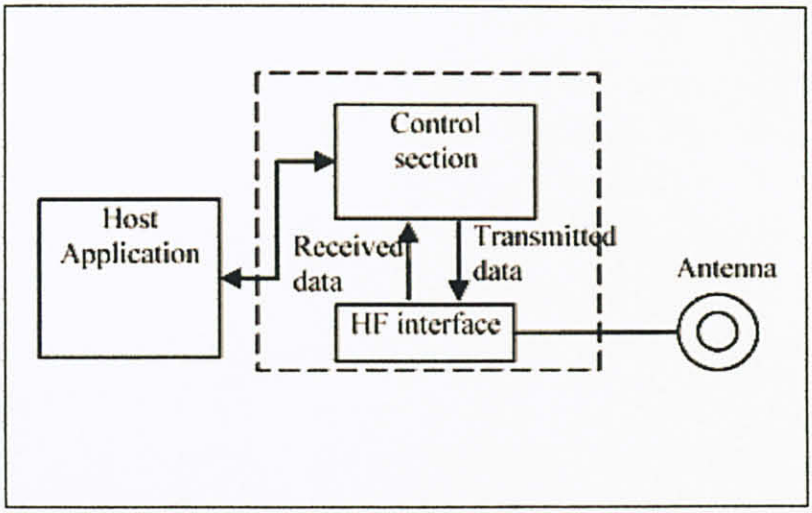


Figure 1: Block diagram of a typical RFID reader.

2.1.2 RFID Transponder or Tag.

RFID tags can either be active, passive or semi-passive.

- *Passive Tag System* does not contain an internal power source. When radio waves are encountered, an electrical current is induced in the antenna to form a magnetic field. This power is what is used to power up the passive RFID tag. Thus, they can therefore only operate in the presence of a reader. The communication range is limited; however it has a much longer lifecycle as it does not require a continuous power source.[2]

- *Active Tag System* has an active radio frequency (RF) transmitter, and requires an internal power source to power the integrated logic circuit and to communicate with the reader. Read range increases (up to several kilometers) and improve reliability. They also have a much larger memory than passive tags and, due to their higher processing capabilities, are also more secure.[2]

- *Semi-passive Tag System* requires a battery to maintain memory in the tag or power the electronics that enable the tag to modulate the reflected signal. However, communicates in the same method as the other passive tags. [2]

To distinguish tag types from each other, Electronic Product Code (EPC) Global has established five tag classes to indicate capabilities a tag can perform. For instance, Class 0 tags are factory programmable. The EPC number is encoded onto those tags during manufacture and can be read by a reader. Class 1 tags can be programmed by the retailer and supplier. They are manufactured without the EPC number which can be encoded onto the tag later in the field (i.e., by retailer and supplier). The Class 3 tags have the Class 2 capabilities plus a power source to provide increased range or advanced functionality. The Class 4 tags have the Class 3 capabilities plus active communication and the ability to communicate with other active tags. The Class 5 tags have the Class 4 capabilities plus the ability to communicate with passive tags as well. [10]

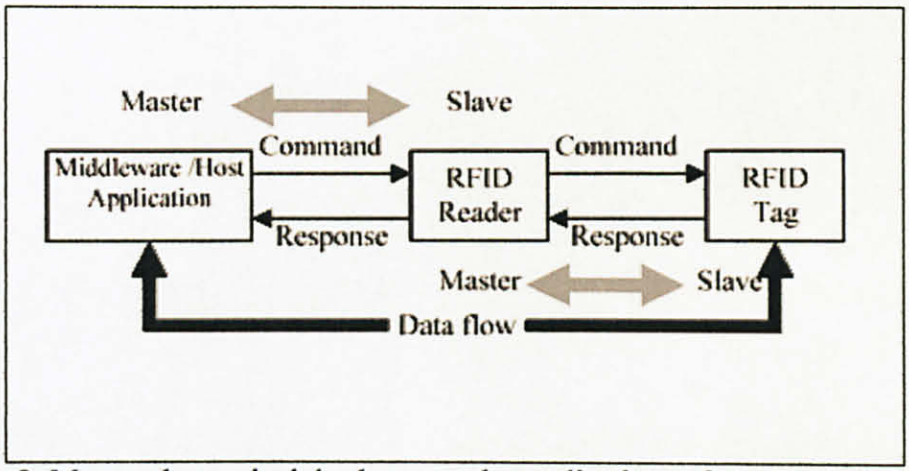


Figure 2: Master-slave principles between the application software and reader, and the reader and transponders

2.1.3 Frequency

Table 1: RFID operating frequencies and associated characteristics

Band	**LF** **Low Frequency**	**HF** **High Frequency**	**UHF** **Ultra High Frequency**
Frequency	30–300kHz	3–30MHz	300 MHz–3GHz
Typical RFID Frequencies	125–134 kHz	13.56 MHz	433 MHz or 865 – 956MHz 2.45 GHz
Approximate read range	less than 0.5 meter	Up to 1.5 meters	433 MHz = up to 100 meters 865-956 MHz = 0.5 to 5 meters
Typical data transfer rate	less than 1 kilobit per second (kbit/s)	Approximately 25 kbit/s	433–956 = 30 kbit/s 2.45 =100 kbit/s
Characteristics	Short-range, low data Transfer rate, penetrates water but not metal.	Higher ranges, reasonable data rate (similar to GSM phone), penetrates water but not metal.	Long ranges, high data transfer rate, concurrent read of <100 items, cannot penetrate water or metals
Typical use	Animal ID, Car immobilizer, Access Control	Library Books, Airline Baggage	Specialist animal Tracking, Parking Lot Access

Frequency refers to the size of the radio waves used to communicate between the RFID systems components. It is generally safe to assume that a higher frequency equates to a faster data transfer rate and longer read ranges, but also more sensitivity to environmental factors such as liquid and metal that can interfere with radio waves. RFID systems currently operate in the Low Frequency (LF), High Frequency (HF) and Ultra High Frequency (UHF) bands.[1]

Each frequency has advantages and disadvantages relative to its capabilities. Generally a lower frequency means a lower read range and slower data read rate, but increased capabilities for reading near or on metal or liquid surfaces.[1]

2.1.4 Memory

Tags come in a variety of forms with varying types of on-chip memory capability. Tags can be read-only (the unique ID code is permanently stored on the tag), read/write (allowing a user to change the ID and add additional data to the tag's memory), or they can be a combination, with a permanent tag ID and some storage space for the user's data.

Passive tags typically have anywhere from 64 bits to 1 kilobyte of non-volatile memory. Active tags tend to have larger memories with a range of, typically, between 16 bytes and 128 kilobytes.[1]

2.2 RFID Solutions for Healthcare Industry

The medical care industry has never been more focused than now on ways to reduce expenses and save more patients lives. Technology is helping to increase these efficiencies. RFID (Radio Frequency Identification) technology specifically is being put to work in different areas of the system such as asset, patient and baby tracking, patient-bedside care, and helping make Operating Room procedures safer. [5]

2.2.1 Advantages of the RFID Technology

There is a high investment in the development and improvement of the RFID systems because of the important advantages that organizations can reach with it when compared with bar code tags where the reading must be done by a visual contact using optical readers. In healthcare particularly, RFID is considered generally more suitable than barcoding and has many potential advantages such as field reading, as opposed to line-of-sight reading. RFID devices can store more data than barcodes and some RFID tags can have data written to them by the interrogator.[11]

2.2.2 Patient Tracking Solution:

Hospitals and Health Care facilities are more concerned these days about having the ability to accurately track and accurately identify their patients. Whether it's dispensing medication or identifying the individual for procedures or pulling up a history of treatment on a patient; RFID technology with the appropriate software is a way to manage the information much more efficiently.

An RFID enabled system is used for accurately locating and tracking people, equipment and objects in a hospital. The system can be designed to track hospital assets, curb excess expenditures, increase safety and ensure security and access control by placing small radio frequency identification devices, or 'RFID tags' on people and objects.

Pinnacle Health in Central Pennsylvania had implemented RFID tracking for patients and medical equipment.[6] Hospitals and Emergency Medical Services (EMS) are facing daily challenges especially when responding to emergencies. They need to be able to manage a large number of patients during an emergency effectively and efficiently.

The institution is responsible for the care and safety of patients. If a patient is missing from their hospital bed or emergency ward the responsibility rests with the healthcare organization. RFID makes tracking and finding patients easier. Administering bedside care is safer, faster, because of realtime and accurate identification and verification of patient.

- RFID bracelets can be worn by patients. An RFID bracelet can be read through bed linens, so patients don't have to be disturbed when sleeping.
- Patients can be geo-fenced by using RFID bracelets and fixed readers in certain doorways and corridors to ensure patients do not stray beyond a predetermined perimeter.
- RFID bracelets can contain some patient information or all the information can reside on a computer database logically linked to the RFID bracelet. [5]

2.2.3 Patient Identification.

Many health professionals are concerned about the growing number of patients who are misidentified before, during or after medical treatment. Indeed, patient identification error may lead to improper dosage of medication to patient, as well as having invasive procedure done. Other related patient identification errors could lead to inaccurate lab work and results reported for the wrong person, having effects such as misdiagnoses and serious medication errors. [7]

In order to cut these clinical errors, to improve patient care and security and also to improve administration and productivity, several RFID-based patient identification and tracking pilot projects have been launched during the last two years. For instance, in New York's Jacobi Medical Center, in the Birmingham Heartlands Hospital or in the German SaarbrÄucken Clinic Winterberg. Concretely, all patients admitted to the hospital are given an RFID based wristband resembling a watch with a passive RFID chip in it. This chip stores a unique patient ID number and

some relevant medical information such as the patient's blood type, in order to speed treatment. To ensure patient privacy and to avoid that medical records are improperly disclosed, further medical data are not stored on the devices but are rather stored in a secure database that links the unique patient's ID with its data. [4]

2.2.4 Newborn Baby Tracking Solutions

The Maternity Ward is a place for joy and celebration as new family members are welcomed into the world. The ward staff shares some of this joy by playing their part throughout the labor and the post-partum care of mother and newborn child. However anxiety levels rise for the patients, families and staff when there are complications during the labor and/or when the newborn needs neonatal care. In many of these instances the baby and the mother need to be separated, a scenario that creates anxiety. Anxiety can easily turn to anger and a possible law suit when there is a mother-baby mix up; the wrong baby being sent to the wrong mother for the crucial early bonding period, breast-feeding.

RFID technology will help both these difficult and challenging situations. At the time of admittance the mother is provided with a patient's RFID tag that will start the process of tracking all procedures relevant to that labor. Once the child is born, an ankle tag is provided to the baby as well and immediately cross-referenced to the baby's mother and the mother's tag. [5]

2.2.5 Family Access to Babies in Neonatal Care:

In the case of the newborn needing neonatal care, family members are almost as anxious as the new mother, to greet and bond with the new family member. When the newborn needs to be in an intensive care unit (ICU) this becomes very difficult due to the restricted access in the ICU. RFID solves this problem by making available to families the ability to view on a computer monitor outside of the ICU the key data

about the baby, a photograph, the height, weight, skin color and temperature after keying in, for example, the mother's family name and first name. [5]

2.2.6 Mother-Baby Mix-ups Eliminated:

Even when the mix up is uncovered moments after the mistake is made, this can cause heightened levels of anger and dissatisfaction with the services rendered. Any mother-baby mix up is a potential liability to the hospital. This scenario can be eliminated when staff follows a procedure of reading the baby's RFID tag and then reading the mother's RFID tag and re-confirming the match that was initiated at the birth. [5]

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

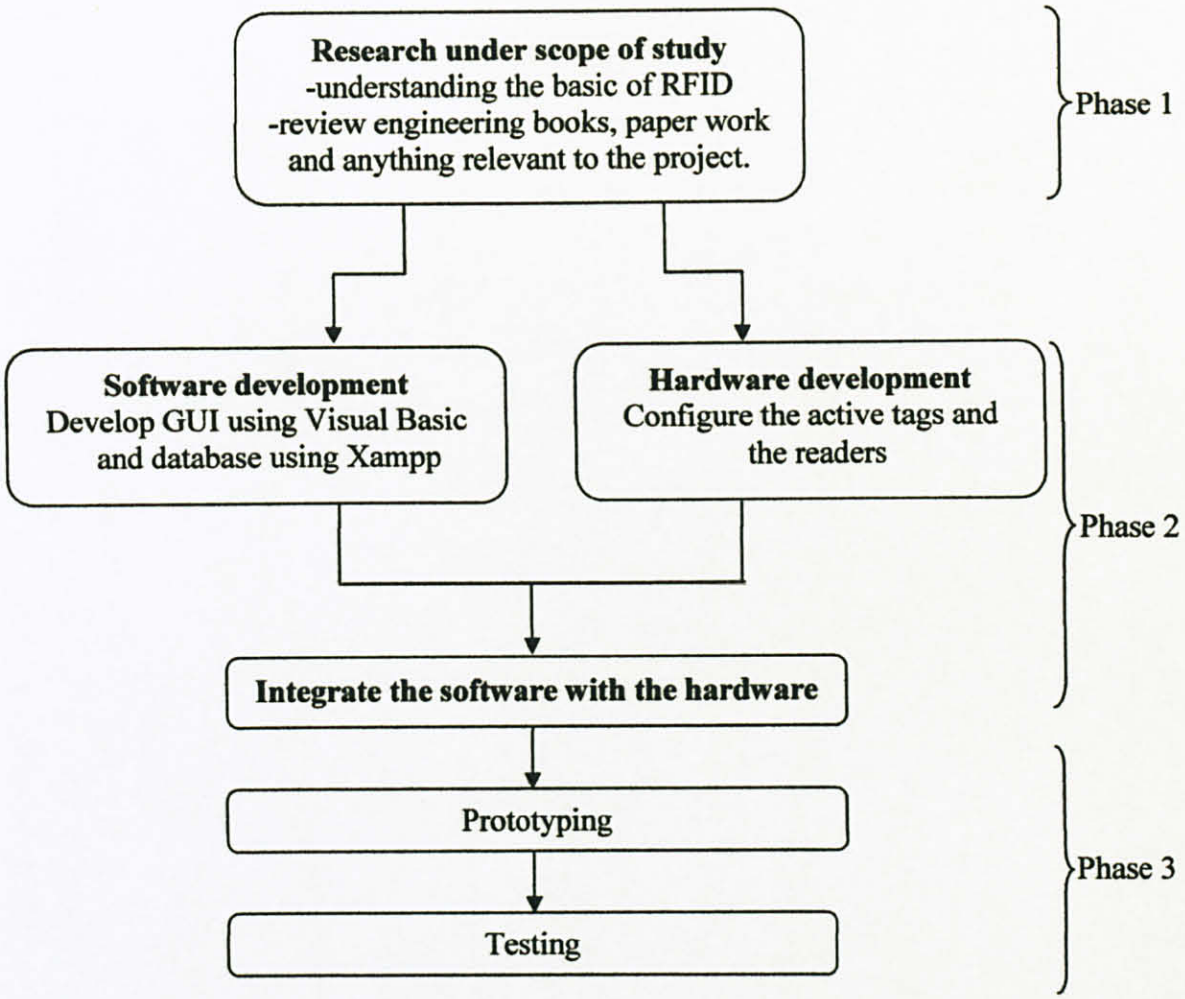


Figure 3 : Flow Chart of the Methodology

3.2 Project Activities

Developing software as an interface between the hardware and the database in a computer is the step that is required in completing this project. Develop the database in the notebook using Xampp-win32 software is the first progress of developing the software. MySQL connector software is to intergrate the interface using Visual Basic with the database from Xampp-win32 software. Understanding the Visual Basic from various books that provide information and guideline to use Visual Basic.

The Author has learnt how to setup the hardware with the software in the host computer to ensure the hardwares are functioning well. Hardware setup is required to ensure the hardware can communicate with the host computer. The Author went to the manufacturer since the host computer could not connect with the hardware during the setup hardware.

3.3 System Overview

These are the steps that are required to ensure the system work smoothly:

1. Exactly after delivering the baby, the RFID bracelet must be attached to the mother's hand and her newborn baby's ankle.
2. The reader that is fixed or mounted on the ceiling will monitor the baby's tag.
3. If the baby's tag matched with mother's tag, it is in a safe situation.
4. If the tag has been tampered or baby is out of the range, the system will activate the alert window.

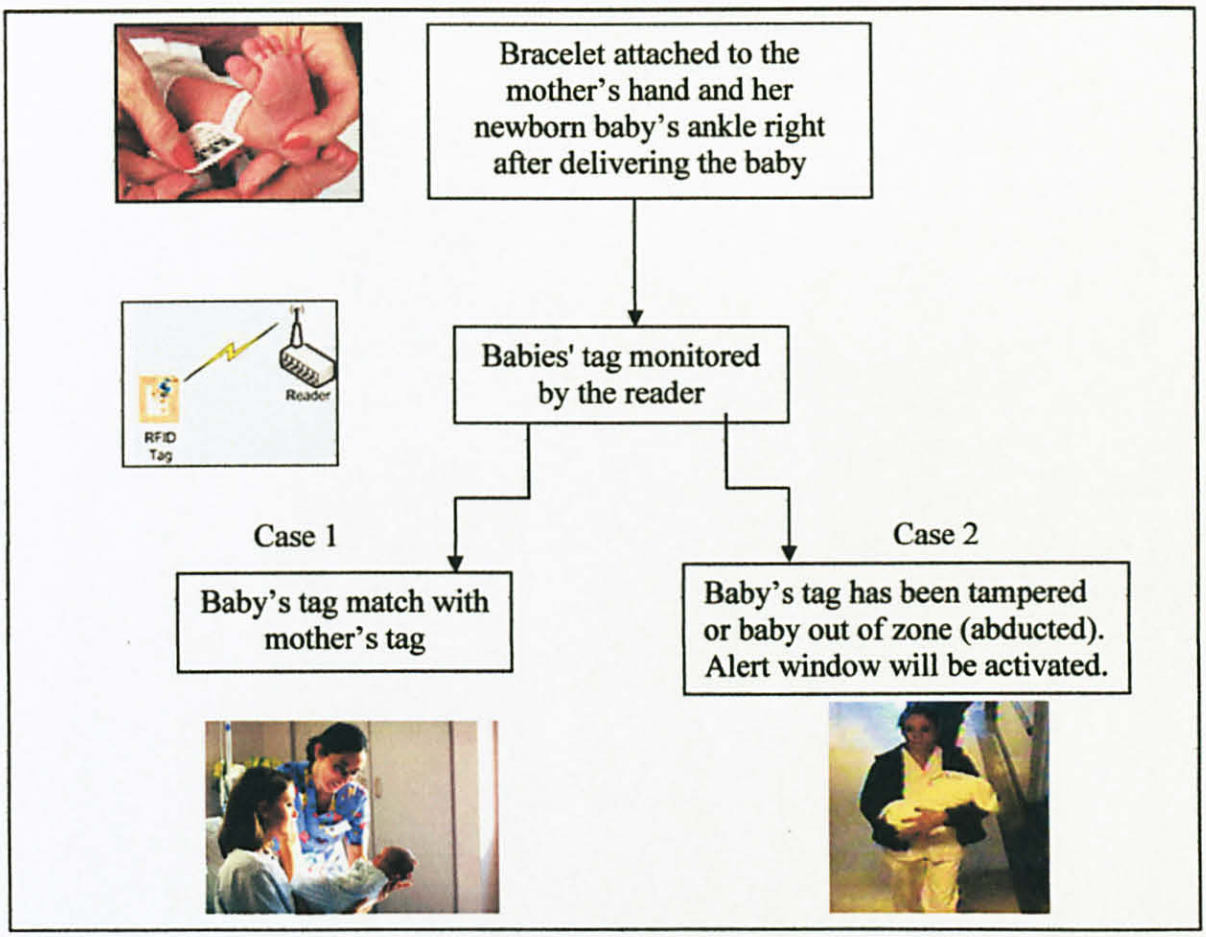


Figure 4: System overview

3.3.1 Basic RFID System.

There are three components in the RFID system which is the host computer, the reader and the tags. Configuration of the reader and the tags are done via either an RS-232 connection of Ethernet connection. The connection from the host computer to the reader will monitor and track the tags. The ActiveWave Reader RS-232 connector cable provides serial communications between the ActiveWave reader and the host computer. Figure 3 shows the basic operation of the RFID system.

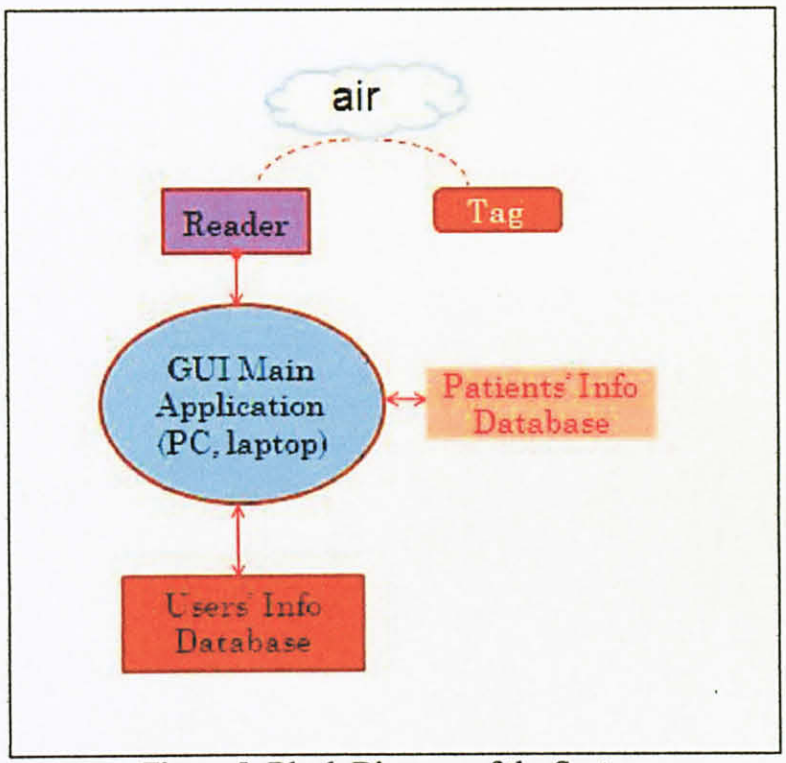


Figure 5: Block Diagram of the System

For the block diagram of the system shown in the Figure 3, users' information and patients' information in the database and the Graphic User Interface (GUI) can be access through the host computer which is in notebook or computer. The host computer and the readers are connected using Ethernet cable shown in the Figure 4. The reader has an antenna that emits radio waves and the tag will responds by sending back its data through air.

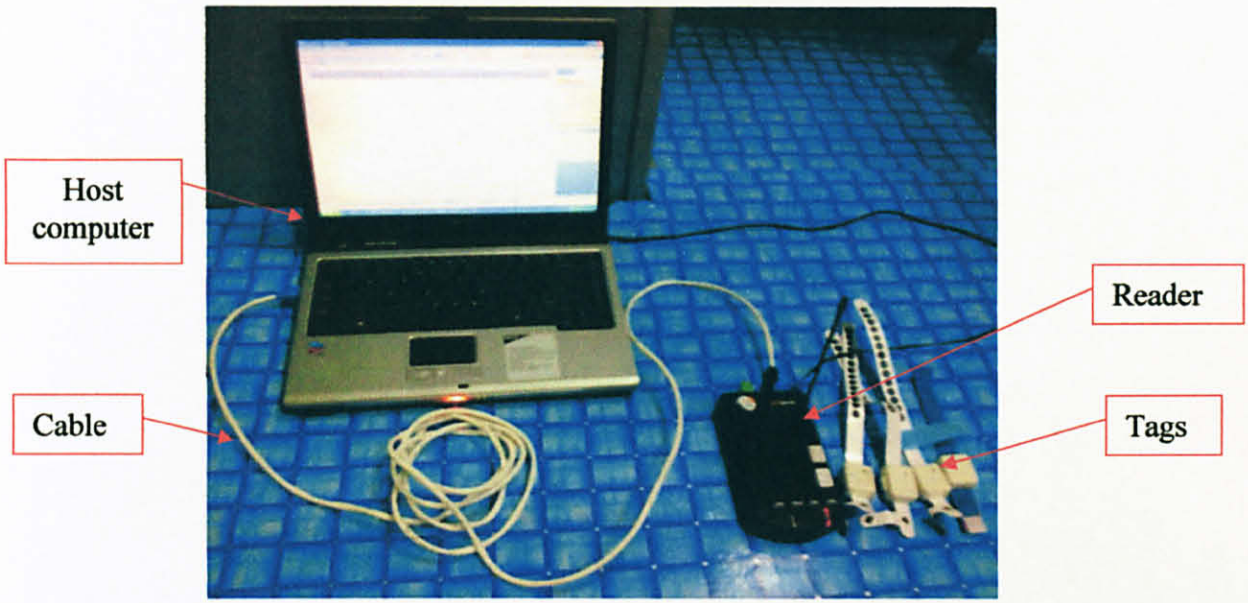


Figure 6: Configuration

3.4 Tools and Equipments Required

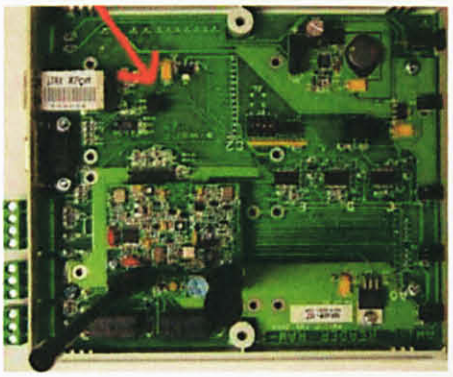
List of software and hardware tools required to develop the system:

Figure 7: RFID Kit

RFID Kit:

- 1 ActiveWave Reader and Power Supply
- 1 ActiveWave Reader RS-232 Connector Cable
- 3 ActiveWave Card Tags
- 2 ActiveWave Wristband Tags
- 2 ActiveWave Jumbo Tags
- 1 ActiveWave Field Generator and Power Supply
- 1 Programming Station Software Application
- 1 Tracker Program Software Application
- 1 API with Documentation and Example Software Application

Table 2: Equipments required and the descriptions

Equipments	Descriptions
RFID Activewave wristband tag 	• Transmitting at 868 MHz • Can initiate communication by sending its own signal. • All ActiveWave tags have anti-collision circuitry to assure each tag's information is received when more than one tag is transmitting.
ActiveWave Reader RS-232 	• Use dual frequency (receive data at 868 MHz and transmit at 433 MHz) • Use beacon mode to enable the signal/data to be sent continuously. • Communicate to the host computer via an RS-232 cable or via an Ethernet network connection. • Used to enable, disable, wake up and program tags.
MySQL Connector Net 5.0.9	• Connector between Microsoft Visual Basic 2008 and Xampp Software.
Xampp Software	• To save the database that contains the information of the patients.
Microsoft Visual Studio Basic 2008	• Use to communicate between hardware and software and as interface to user.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Graphical User Interface (GUI) of RFID Tagging System for Newborn Babies.

This is how the system for the Graphical User Interface (GUI) works using Visual Basic (VB). Firstly, the user or the hospital's staff has to connect the reader with the system by open the RFID Query Server window. The users have to click at the 'Scan Network' button to identify active hosts on a network. Follow by clicking a 'Open Socket' button to specify a particular port number to use. Then, the 'Query tag' button is to respond all the tags to sending back their data or signal to the reader.

Figure 8: RFID Query Server

For security purpose,the user has to login the system by insert the registered user name and password in the Login window as shown below:

Figure 9: Login Window

After successfully login into the system, the Menu Window will appear that contain "Start" button, "Add Baby" button, " Baby List" button, "Add User" button "User List" button,"Log Out" button and also an 'Alert' and 'Match' window in the Menu window as shown in Figure 10.

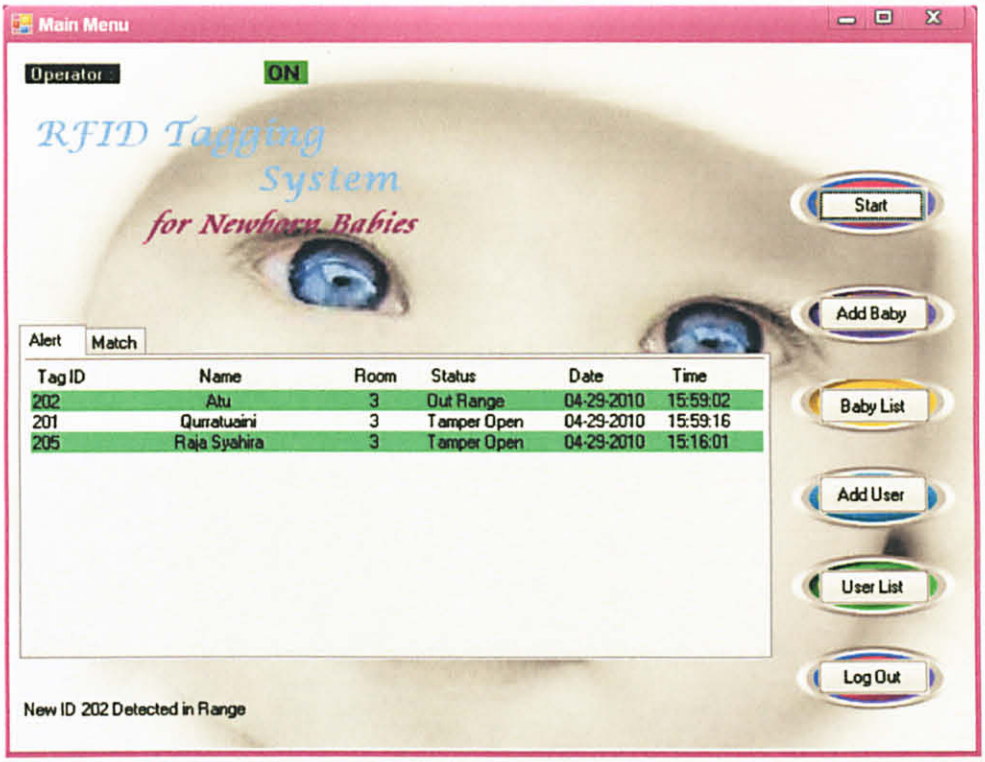


Figure 10: Menu Window

Firstly, to connect the system, the user has to click at the "Start" button. The "OFF" sign next to the start button will change to "ON" sign at the Menu Screen showing that the system is connected. All the tags will be monitored in order to update the status of the babies' and mothers' tags.

To add patient, the user has to click "Add Baby" button from the Menu screen and the "Add New Baby" screen will pop out. The "Add New Baby" window as shown below:

Figure 11: Add Patient Window

The user has to fill up all the information of the mother and her baby and make sure both of the tags ID are match before attach the bracelet to the mother's hand and her baby's ankle. The user has to click "Submit" button after complete fill up the form and the information will send to the database.

To add the new user to control the system, the user has to click "Add User" button at the "Menu" window and the "Add User" window will pop out as shown below:

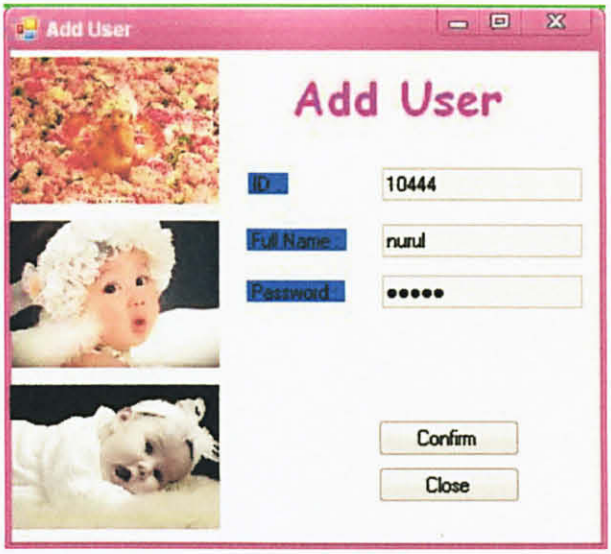


Figure 12: Add New User Window

When the user has fill up the information in the "Add User" window, click "Confirm" button and all the information will key in into the database as in Figure 14. To check the users list, the user has to click "User List" button at the "Menu" window and the "Personnel List" window will appear as shown in Figure 13.

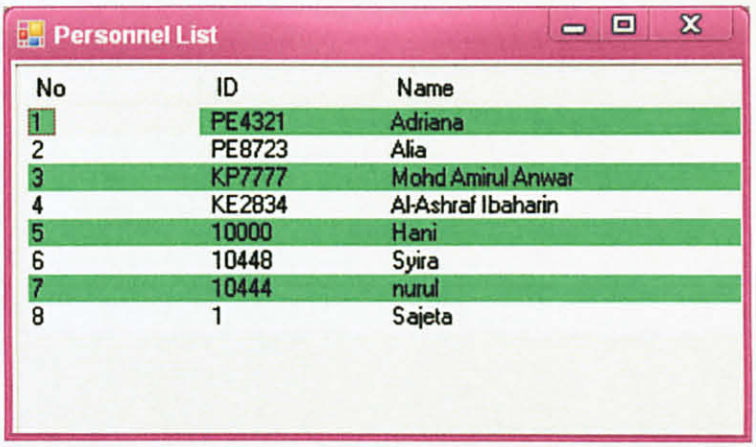


Figure 13: List of Users

+ Options

	id	name	Password
	PE4321	Adriana	4321
	PE8723	Alia	8723
	KP7777	Mohd Amirul Anwar	7777
	KE2834	Al-Ashraf Ibaharin	2834
	10000	Hani	10000
	10448	Syira	10448
	10444	nurul	10444
	1	Sajeta	1

Check All / Uncheck All *With selected:*

Figure 14: Database Staffs' Informations

To check the patients list, the user has to click "Baby List" button at the "Menu" window and the "Baby List" window will appear as shown in Figure 15. All the informations are stored in Xampp database as shown in Figure 16.

Figure 15: List of Patients

+ Options

	name	Room	mother	tag_m	tag_id	P_Date	P_Time
	Qurratuaini	3	Atu	202	201	10/4/2010	11:30
	Raja Syahira	3	Radja	205	203	23/7/2010	18:30
	Nisha Saharudin	3	Bedah	1000	1001	13/2/2010	20:18

Check All / Uncheck All *With selected:*

Figure 16: Database List of Patients

To detect the tag either it is out of zone or has been tampered, the system will monitor all the tags after the user click a 'Start' button at the Menu window and if one of the tag is out of area, the warning window will pop out mention that the tag is out of zone as Figure 17. The alert window will show the name of the baby , the tag ID of the baby, room number and the time when the tag is out of zone.

Figure 17: Tag out of zone

The system also can monitor if the baby tag has been tampered. If someone try to remove the tag from the baby without hospital permission,the warning window will pop out as Figure 18. If the user close the tag out of zone and tag tempered window, the user still can observe the tag at the 'Alert' window in the Menu window. The 'Alert' window will show the update of the tag either it has been tempered or it is out of zone.

Figure 18: Tag tampered

To match the mother's tag and baby's tag, ensure both of the tags are in the reader zone. The tag ID of the mother and the baby will be crosschecked to make sure the baby belongs to his/her respective mother. To monitor the matching process, the user has to click 'Match' window in the Menu window to observe both the mother's tag and her baby's tag is match or mismatch. The red color represent both tags are not matched and the green color represent both mother's and baby's tag are matched as shown below:

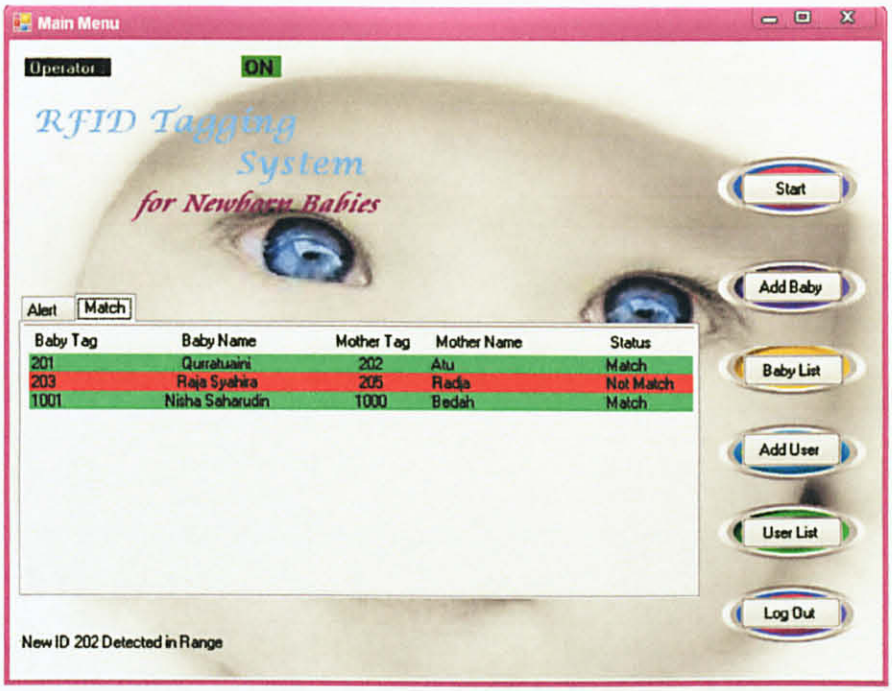


Figure 19: Matching window

4.2 Interface ActiveWave of RFID Tagging System for Newborn Babies.

The hardwares which are the tags and the readers has been tested to make sure they are functioning well. Hardware setup is required to ensure the hardware can communicate with the host computer. This is how the host computer communicates with the hardware which is RFID using ActiveWave. Firstly, the user has to access ActiveWave and the Communication Configuration Dialog window will appeared as shown below:

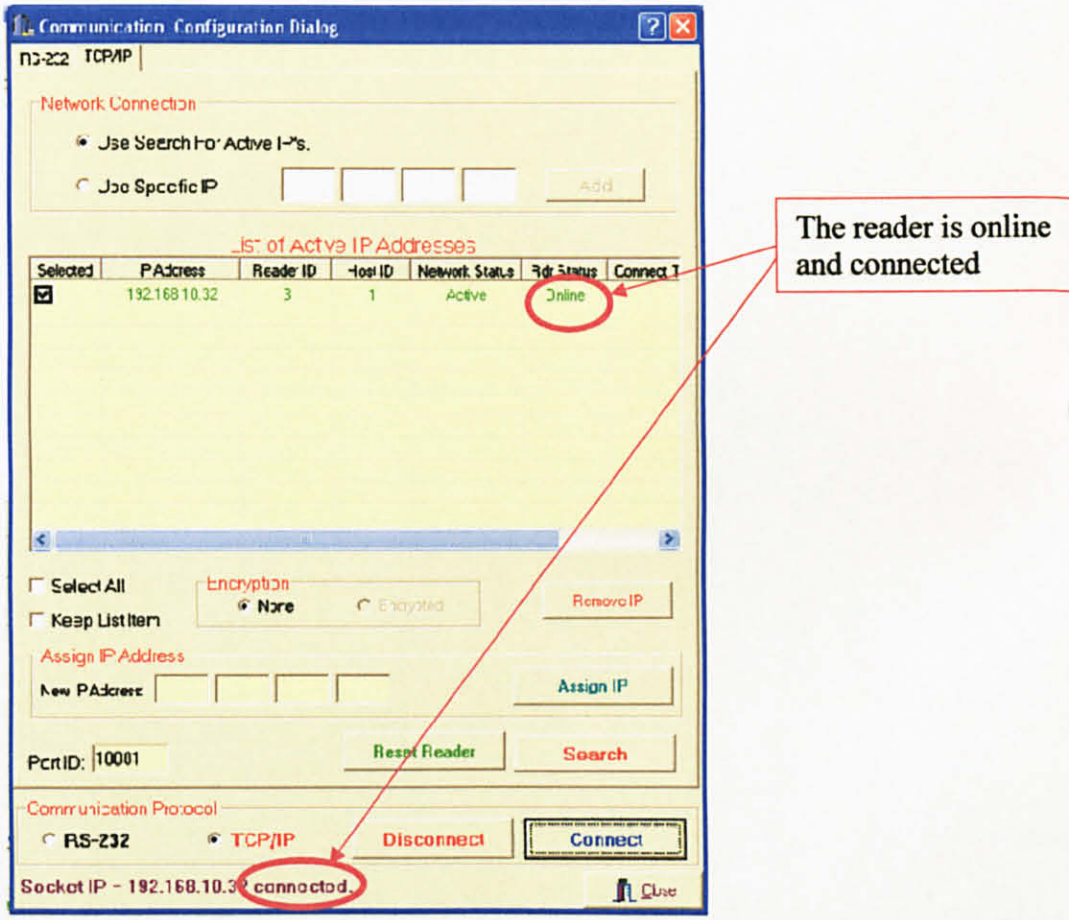


Figure 20: Communication Configuration Dialog window

To communicate with the reader, the user has to click 'Search' and the reader's address will be appeared. To ensure the IP Address in the Communication Configuration Dialog window is belongs to the specific reader, the user cans double check the address at the reader. To connect to the reader, the user has to click 'Connect' and the reader status will show 'online' if the reader is successfully communicate with the host computer.

The Programming Station window will show the tags that detected after the reader emits the radio waves as shown in the Figure 21. The tags can be identified by the specific ID that appeared. All of the active tags have anti-collision circuitry that assures each tag's information is received when more than one tag is transmitting.

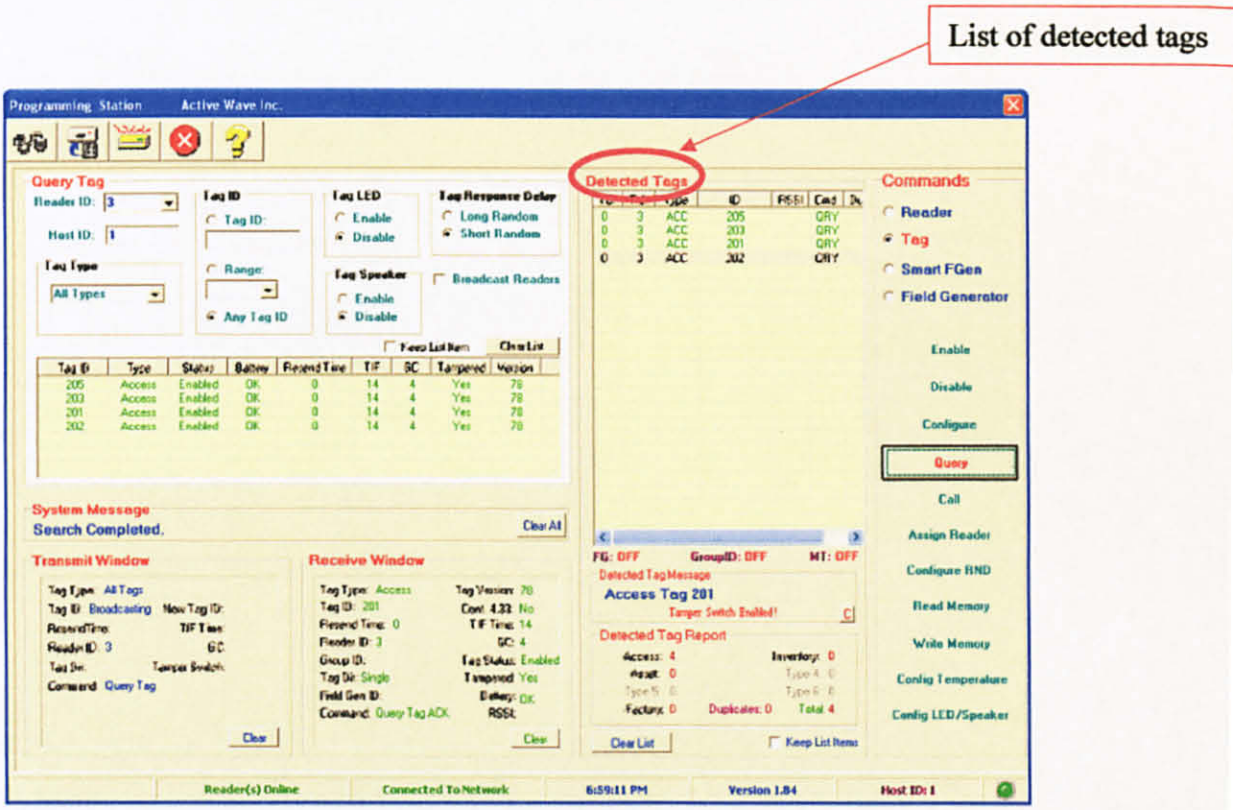


Figure 21: Programming Station

4.3 Discussion

During the development of the project, the author had faced some problem to learn the Visual Basic 2008 and Xampp software since the author had never been exposed to both of the software. This process really took alot of finding information, theoretical studies and consultation from the experts.

Hardware setup is required to ensure the hardware can communicate with the host computer. The Author went to the manufacturer since the host computer could not connect with the hardware during the setup hardware. Another problem encountered during the development of the project was limited access to the hardware since there are quite a number of students using the same hardware due to the limited budget and cost of the hardware. On the other hand, since this project is using a RFID active wristband tag, higher consumption of battery usage is needed due to many application of the tag.

Despite of the difficulties, the project still managed to be accomplished successfully according to plan.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Using Radio Frequency Identification (RFID) Tagging System for Newborn Babies really can keep away babies from abduction and mismatching problem. This idea of this project can make the hospital handling newborn flow smoothly until handling over to their respected parents and improve security features. The system will match RFID tag between mother and her newborn baby. If the baby is given to a wrong mother, the RFID tags will alert the mismatch and activate an audible alarm. On the other hand, the straps use a tamper proof strap that can avoid someone from tamper the strap identification and if the strap has been tampered, the system will trigger an alarm to alert the hospital security to this situation. The system will also trigger an alarm if the baby's strap is out of range which means the baby has been abducted. The working prototype has been tested and the testing has been done successfully.

5.2 Recommendation

There are some recommendations that can be made to improve this project. Firstly, a GSM can be implementing for the system to improve the security status by sending SMS directly to the authorized person. The basic operation for this project there will be a pop out window if the tag has been tampered or the tag is out of zone. By adding a GSM, an authorized person can be always updating the status of the babies. On the other hand, for the future improvement of the project, body temperature sensor can be implementing to detect babies' temperature. Beside to monitor the tampered and out of zone matter, the system also can detect the level temperature of the babies.

REFERENCES

[1] Gaynor Backhouse, JISC TechWatch, May 2006 "*RFID: Frequency, standards, adoption and innovation*", JISC Technology and Standards Watch.

[2] Elisabeth Ilie-Zudor, 2006 "*The RFID Technology And Its Current Applications*" Computer and Automation Research Institute, Hungarian Academy of Sciences.

[3] Stevan Preradovic, Nemai C. Karmakar, "*RFID Readers - A Review*", 4th International Conference on Electrical and Computer Engineering ICECE 2006, 19-21 December 2006, Dhaka, Bangladesh, pp. 96-99

[4] Patrik Fuhrer, Dominique Guinard, "*Building a Smart Hospital using RFID technologies*", University of Fribourg, Department of Informatics.

[5] GAO RFID Inc.< http://healthcare.gaorfid.com/ >

[6] Radianse <http://www.radianse.com/success-stories-pinnacle.html>

[7] P. Sirivattha. 2003, Issues on patient identification related error. Technical report, University of Hawaiiat Manoa, Information and Computer Sciences Department.

[8] Stephen Smith, Globe Staff , "*Breast-feeding mixup at hospital, Mother given wrong newborn*", By October 26, 2004, Copyright 2006 Globe Newspaper Company.

[9] Atkinson, W. (2004) 'Tagged: the risks and rewards of RFID technology', *Risk Management*, Vol. 51, No. 7. pp. 12-18.

[10] Meyerson, J. (2007) *RFID in supply Chain: A Guide to Selection and Implementation*, Taylor & Francisc Group, New York.

[11] Symonds, J., Parry, D. and Briggs, J. (2007) 'An RFID-based System for Assisted Living: Challenges and Solutions'. *Studies in Health Technology and Informatics*, Vol. 127, pp.127- 38.

APPENDICES

APPENDIX A

Coding Login window in VB

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports System.IO
Imports MySql.Data.MySqlClient


Public Class officer

    Public ofName As String

    Private Sub officer_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Function checkPswd(ByVal name As String, ByVal pswd As String) As Boolean

        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="
        Dim cn As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM officer WHERE id = '" & name & "'AND Password = '" &
pswd & "'", cn)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable
        Dim offName As String = ""

        cn.Open()

        Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "employees")
            dtGH = dsGH.Tables("employees")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If

        If (rows > 0) Then
            found = True
            Dim namCmd As New MySqlCommand("SELECT name FROM officer WHERE id = '" & name & "'", cn)
            Dim rdr As MySqlDataReader
            cn.Open()
            rdr = namCmd.ExecuteReader
            While rdr.Read
                offName = rdr.Item("name").ToString
            End While
```

```
            rdr.Close()
            cn.Close()
            ofName = offName
        Else
            found = False
        End If

        Return found

    End Function

    Private Sub LogoPictureBox_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub lbl_Password_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
lbl_Password.Click

    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Me.Close()
    End Sub

    Private Sub btn_Submit_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_Submit.Click

        If (checkPswd(Me.TxtName.Text, Me.TxtPswd.Text) = True) Then
            main.Show()
            Me.Hide()
        Else
            MessageBox.Show("Invalid Name or Password")
        End If

    End Sub

    Private Sub TxtName_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TxtName.TextChanged

    End Sub

    Private Sub L_Welcome_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Login.Click

    End Sub

    Private Sub UsernameLabel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
UsernameLabel.Click

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_clear.Click
        TxtName.Text = ("")
        TxtPswd.Text = ("")
    End Sub
End Class
```

Coding Menu window in VB

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class main

  Dim Hconn As IntPtr
  Dim readerIP(20) As Byte
  Dim readerPort As UInt16
  Dim commPort As UInt32
  Dim commBaud As UInt32
  Dim myPKTID As Integer
  Dim registered As Boolean
  Dim strhttp As String


  Dim ipIdx As Integer = 0
  Dim pubReaderID As Integer
  Dim c As Integer
  Dim txtSender As String

  Dim tagID As ListViewItem
  Dim lsview As ListViewItem
  Private Strt As System.Threading.Thread
  Dim strCn As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="
  Dim selID As String


  Private Sub main_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load


    registered = False


    'Strt.Start()

  End Sub

  Private Sub InsertRow(ByVal tgid As Integer, ByVal st As Integer, ByVal rdr As Integer, ByVal tpr As String)

    Dim name As String = ""
    Dim zone As String = ""
    Dim location As String = ""
    Dim rowsAlarm As Integer
    Dim dat As DateTime = Nothing
    Dim id As String = ""

    Dim cn As New MySqlConnection(strCn)
    ' If the connection string is null, use a default.
    'MsgBox("Start")
    'Beep()

    If st = 0 Then


      cn.Open()

      Dim cmdName As New MySqlCommand("SELECT * FROM patients WHERE tag_id = " & tgid, cn)
      Dim readerInmate As MySqlDataReader
```

```
readerInmate = cmdName.ExecuteReader()
While readerInmate.Read
   id = readerInmate.Item("id")
   name = readerInmate.Item("name")
   zone = readerInmate.Item("Room")
End While
readerInmate.Close()
cn.Close()


cn.Open()
Dim cmdLoc As New MySqlCommand("SELECT RoomNo FROM readers WHERE ReaderID=" & rdr, cn)
Dim readerLoc As MySqlDataReader
readerLoc = cmdLoc.ExecuteReader()
While readerLoc.Read
   location = readerLoc.GetString(0)
End While
readerLoc.Close()
cn.Close()


Dim cmdAlarm As New MySqlCommand("SELECT * FROM alarm WHERE id = '" & id & "'", cn)
Dim daAlarm As New MySqlDataAdapter
Dim dsAlarm As New DataSet
Dim dtAlarm As New DataTable

cn.Open()

Try
   daAlarm.SelectCommand = cmdAlarm
   daAlarm.Fill(dsAlarm, "searchresult")
   dtAlarm = dsAlarm.Tables("searchresult")
   rowsAlarm = dtAlarm.Rows.Count()
Catch ex As Exception
   MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
End Try

If ConnectionState.Open Then
   cn.Close()
End If

If tpr = "False" Then

   If rowsAlarm = 0 Then

      If location = zone Then
         cn.Open()
         Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack, id, Name, Zone, ProgressTime) VALUES
('OK' " & ",'" & id & "','" & name & "'," & location & ",'" & Format(DateTime.Now(), "yyyy-MM-dd hh:mm:ss") & "')", cn)
         cmdIns1.ExecuteNonQuery()
         cn.Close()
      Else
         cn.Open()
         Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack, id, Name, alertType, Zone, ProgressTime)
VALUES ('Alert' " & ",'" & id & "','" & name & "', 'OUT OF ZONE', " & location & ",'" & Format(DateTime.Now(), "yyyy-
MM-dd hh:mm:ss") & "')", cn)
         cmdIns1.ExecuteNonQuery()
         cn.Close()
         'MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" + name + " moved from Zone" + zone.ToString + " To
Zone" + location.ToString)
         MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" + name + " is out of zone!!")
      End If


   Else

      If location = zone Then
```

```
                cn.Open()
                Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack = 'OK', alertType = '', ProgressTime = '"
& Format(DateTime.Now, "yyyy-MM-dd hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id & "'", cn)
                cmdUpdate2.ExecuteNonQuery()
                If ConnectionState.Open Then
                  cn.Close()
                End If
              Else
                cn.Open()
                Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack = 'Alert', alertType = 'OUT OF ZONE',
ProgressTime = '" & Format(DateTime.Now, "yyyy-MM-dd hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id &
"'", cn)
                cmdUpdate2.ExecuteNonQuery()
                cn.Close()
                'MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" + name + " moved from Zone" + zone.ToString + " To
Zone" + location.ToString)
                MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" + name + " is out of zone!!")
              End If

            End If

          ElseIf tpr = "True" Then
            'alert.lblType.Text = "TEMPERED!!!!"
            'alert.txtTime.Text = Format(DateTime.Now, "hh:MM:ss dd-mm-ss")
            'alert.txtName.Text = name
            'alert.txtLocation.Text = location
            'alert.txtID.Text = id
            If rowsAlarm = 0 Then

              cn.Open()
              Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack, id, Name, alertType, Zone, ProgressTime)
VALUES ('ALERT' " & ",'" & id & "','" & name & "', 'TEMPERED'," & location & ",'" & Format(DateTime.Now(), "yyyy-
MM-dd hh:mm:ss") & "')", cn)
              cmdIns1.ExecuteNonQuery()
              cn.Close()

            Else

              cn.Open()
              Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack = 'ALERT', alertType = 'TAMPERED',
ProgressTime = '" & Format(DateTime.Now, "yyyy-MM-dd hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id &
"'", cn)
              cmdUpdate2.ExecuteNonQuery()
              cn.Close()

            End If
            MessageBox.Show("Alert!!!! (Tag ID:" + tgid.ToString + ")" + name + " is Tempered")

          End If
        End If

        ' Strt.Start()

    End Sub


    Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnConnect.Click

        commStatus.BackColor = Color.DarkSeaGreen
        'activeReader()
        BtnStart.Enabled = True
    End Sub



    Private Sub BtnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnStart.Click
```

```
    checkdata()

    Me.commStatus.BackColor = System.Drawing.Color.Green

    Me.commStatus.Text = "ON"
    Timer1.Enabled = True
End Sub
Private Sub checkdata()

    Dim check_id(1000) As String
    Dim check_id_m(1000) As String
    Dim check_name(1000) As String
    Dim check_mother(1000) As String
    Dim check_room(1000) As String
    Dim check_date As String
    Dim check_time As String
    Dim check_status As String
    Dim i As Integer = 0
    Dim j As Integer
    Dim cn As New MySqlConnection(strCn)
    Dim cmd As New MySqlCommand("SELECT * FROM patients", cn)
    Dim da As New MySqlDataAdapter(cmd)
    Dim ds As New DataSet()
    Dim myReader As MySqlDataReader

    'Open Drug database
    If cn.State = ConnectionState.Closed Then
        cn.Open()
    End If
    'read druglist database and store to buffer
    myReader = cmd.ExecuteReader()

    While myReader.Read
        i = i + 1
        check_id(i) = myReader.Item("tag_id").ToString
        check_id_m(i) = myReader.Item("tag_m").ToString
        check_name(i) = myReader.Item("name").ToString
        check_mother(i) = myReader.Item("mother").ToString
        check_room(i) = myReader.Item("Room").ToString

    End While

    If cn.State = ConnectionState.Open Then
        cn.Close()
    End If
    ' complete read druglist

    For j = 1 To i

        ' check data in alarm

        Dim cnchk As New MySqlConnection(strCn)
        Dim cmdchk As New MySqlCommand("SELECT * FROM readers", cnchk)

        Dim dachk As New MySqlDataAdapter(cmdchk)
        Dim dschk As New DataSet()
        Dim myReaderchk As MySqlDataReader
        Dim found As Boolean


        If cnchk.State = ConnectionState.Closed Then
            cnchk.Open()
        End If

        myReaderchk = cmdchk.ExecuteReader()

        While myReaderchk.Read
```

```
            'str = myReader.getString(0)

            If check_id(j) = myReaderchk.Item("ReaderID").ToString Then

               found = True
               Exit While
            Else
               found = False
            End If

         End While

         If found = False Then

            Dim rowscheck As String = 0
            Dim cn2 As New MySqlConnection(strCn)

            Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm  WHERE Alrm_TagId = '" & check_id(j) & "'",
cn2)
            Dim daCheck As New MySqlDataAdapter(cmdCheck)
            Dim dsCheck As New DataSet()
            Dim dtCheck As New DataTable

            If cn2.State = ConnectionState.Closed Then
               cn2.Open()
            End If

            Try
               With daCheck
                  .SelectCommand = cmdCheck
                  .Fill(dsCheck, "searchresult")
               End With
               dtCheck = dsCheck.Tables("searchresult")
               rowscheck = dtCheck.Rows.Count()

               'If unable to connect, show error!
            Catch ex As Exception
               MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
            End Try

            If cn2.State = ConnectionState.Open Then
               cn2.Close()
            End If

            If rowscheck = 0 Then
               cn2.Open()
               check_status = "Out Range"
               check_date = DateString.ToString
               check_time = TimeString.ToString
               'Check for name and if no existing inmates available, add them
               Dim cmdIns As New MySqlCommand("INSERT INTO alarm (Alrm_TagID,
Alrm_Room,Alrm_TagName,Alrm_Status,alrm_Date,Alrm_Time) VALUES ('" & check_id(j) & "','" & check_room(j) & "','"
& check_name(j) & "','" & check_status & "','" & check_date & "','" & check_time & "')", cn2)
               cmdIns.ExecuteNonQuery()
               If cn2.State = ConnectionState.Open Then
                  cn2.Close()
               End If
               'If registration successful, show ID and name
               'Dim style = MsgBoxStyle.Critical

               'MsgBox("!!!!  ALARM !!!!!" & vbCrLf & "Tag ID " & check_id(j) + " Out Of Range Detected",
MsgBoxStyle.Critical, "Alarm")
               alert.lblType.Text = "BABY TAG OUT OF ZONE !!"
               alert.txtID.Text = check_id(j)
               alert.txtName.Text = check_name(j)
               alert.txtLocation.Text = (check_room(j))
```

```
            alert.txtTime.Text = check_time
            ' End If ' if found 2
            alert.Show()

          End If

          'End While
          If cnchk.State = ConnectionState.Open Then
            cnchk.Close()
          End If

        End If ' if found

      Next j
      'TmrCall.Enabled = True
      'displaylistview()
      'removealarm()
      'clearReader()
      checkdata2()
    End Sub
    Private Sub checkdata2()

      Dim check_id(1000) As String
      Dim check_id_m(1000) As String
      Dim check_name(1000) As String
      Dim check_mother(1000) As String
      Dim check_room(1000) As String
      Dim check_date As String
      Dim check_time As String
      Dim check_status As String
      Dim i As Integer = 0
      Dim j As Integer
      Dim cn As New MySqlConnection(strCn)
      Dim cmd As New MySqlCommand("SELECT * FROM patients", cn)
      Dim da As New MySqlDataAdapter(cmd)
      Dim ds As New DataSet()
      Dim myReader As MySqlDataReader

      'Open Drug database
      If cn.State = ConnectionState.Closed Then
        cn.Open()
      End If
      'read druglist database and store to buffer
      myReader = cmd.ExecuteReader()

      While myReader.Read
        i = i + 1
        check_id(i) = myReader.Item("tag_id").ToString
        check_id_m(i) = myReader.Item("tag_m").ToString
        check_name(i) = myReader.Item("name").ToString
        check_mother(i) = myReader.Item("mother").ToString
        check_room(i) = myReader.Item("Room").ToString

      End While

      If cn.State = ConnectionState.Open Then
        cn.Close()
      End If
      ' complete read druglist

      For j = 1 To i

        ' check data in alarm

        Dim cnchk As New MySqlConnection(strCn)
        Dim cmdchk As New MySqlCommand("SELECT * FROM readers", cnchk)
```

```
        Dim dachk As New MySqlDataAdapter(cmdchk)
        Dim dschk As New DataSet()
        Dim myReaderchk As MySqlDataReader
        Dim found As Boolean


        If cnchk.State = ConnectionState.Closed Then
            cnchk.Open()
        End If

        myReaderchk = cmdchk.ExecuteReader()

        While myReaderchk.Read

            'str = myReader.getString(0)

            If check_id_m(j) = myReaderchk.Item("ReaderID").ToString Then

                found = True
                Exit While
            Else
                found = False
            End If

        End While
        If cnchk.State = ConnectionState.Open Then
            cnchk.Close()
        End If



        If found = False Then

            Dim rowscheck As String = 0
            Dim cn2 As New MySqlConnection(strCn)

            Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm  WHERE Alrm_TagId = '" & check_id_m(j) &
"'", cn2)
            Dim daCheck As New MySqlDataAdapter(cmdCheck)
            Dim dsCheck As New DataSet()
            Dim dtCheck As New DataTable

            If cn2.State = ConnectionState.Closed Then
                cn2.Open()
            End If

            Try
                With daCheck
                    .SelectCommand = cmdCheck
                    .Fill(dsCheck, "searchresult")
                End With
                dtCheck = dsCheck.Tables("searchresult")
                rowscheck = dtCheck.Rows.Count()

                'If unable to connect, show error!
            Catch ex As Exception
                MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
            End Try

            If cn2.State = ConnectionState.Open Then
                cn2.Close()
            End If

            If rowscheck = 0 Then
                cn2.Open()
                check_status = "Out Range"
                check_date = DateString.ToString
```

```
            check_time = TimeString.ToString
            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO alarm (Alrm_TagID,
Alrm_Room,Alrm_TagName,Alrm_Status,alrm_Date,Alrm_Time) VALUES ('" & check_id_m(j) & "','" & check_room(j) &
"','" & check_mother(j) & "','" & check_status & "','" & check_date & "','" & check_time & "')", cn2)
            cmdIns.ExecuteNonQuery()
            If cn2.State = ConnectionState.Open Then
              cn2.Close()
            End If
            'If registration successful, show ID and name
            'Dim style = MsgBoxStyle.Critical

            'MsgBox("!!!!  ALARM !!!!!" & vbCrLf & "Tag ID " & check_id(j) + " Out Of Range Detected",
MsgBoxStyle.Critical, "Alarm")
            alert.lblType.Text = "MOTHER TAG OUT OF ZONE !!"
            alert.txtID.Text = check_id_m(j)
            alert.txtName.Text = check_mother(j)
            alert.txtLocation.Text = (check_room(j))
            alert.txtTime.Text = check_time
            '  End If ' if found 2
            alert.Show()

          End If

          'End While


        End If ' if found

      Next j
      TmrCall.Enabled = True
      displaylistview()
      removealarm()
      clearReader()
    End Sub
    Private Sub tamper()

      Dim check_id(1000) As String
      Dim check_id_m(1000) As String
      Dim check_name(1000) As String
      Dim check_mother(1000) As String
      Dim check_room(1000) As String
      Dim check_date As String
      Dim check_time As String
      Dim check_status As String
      Dim i As Integer = 0
      Dim j As Integer
      Dim cn As New MySqlConnection(strCn)
      Dim cmd As New MySqlCommand("SELECT * FROM patients", cn)
      Dim da As New MySqlDataAdapter(cmd)
      Dim ds As New DataSet()
      Dim myReader As MySqlDataReader

      'Open Drug database
      'On Error GoTo errorhandle
      If cn.State = ConnectionState.Closed Then
        cn.Open()
      End If
      'read druglist database and store to buffer
      myReader = cmd.ExecuteReader()

      While myReader.Read
        i = i + 1
        check_id(i) = myReader.Item("tag_id").ToString
        check_id_m(i) = myReader.Item("tag_m").ToString
        check_name(i) = myReader.Item("name").ToString
        check_mother(i) = myReader.Item("mother").ToString
```

```
        check_room(i) = myReader.Item("Room").ToString

    End While

    If cn.State = ConnectionState.Open Then
        cn.Close()
    End If
    ' complete read druglist

    For j = 1 To i

        ' check data in alarm

        Dim cnchk As New MySqlConnection(strCn)
        Dim cmdchk As New MySqlCommand("SELECT * FROM tamper", cnchk)

        Dim dachk As New MySqlDataAdapter(cmdchk)
        Dim dschk As New DataSet()
        Dim myReaderchk As MySqlDataReader
        Dim found As Boolean


        If cnchk.State = ConnectionState.Closed Then
            cnchk.Open()
        End If

        myReaderchk = cmdchk.ExecuteReader()

        While myReaderchk.Read

            'str = myReader.getString(0)

            If check_id(j) = myReaderchk.Item("tpr_tagid").ToString Then

                found = True

                Exit While
            Else
                found = False
            End If

        End While

        If cnchk.State = ConnectionState.Open Then
            cnchk.Close()
        End If



        If found = True Then

            Dim rowscheck As String = 0
            Dim cn2 As New MySqlConnection(strCn)

            Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm  WHERE Alrm_TagId = '" & check_id(j) & "'",
cn2)
            Dim daCheck As New MySqlDataAdapter(cmdCheck)
            Dim dsCheck As New DataSet()
            Dim dtCheck As New DataTable
            'On Error Resume Next
            If cn2.State = ConnectionState.Closed Then

                cn2.Open()
            End If

            Try
                With daCheck
```

```
            .SelectCommand = cmdCheck
            .Fill(dsCheck, "searchresult")
          End With
          dtCheck = dsCheck.Tables("searchresult")
          rowscheck = dtCheck.Rows.Count()

          'If unable to connect, show error!
        Catch ex As Exception
          MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        If cn2.State = ConnectionState.Open Then
          cn2.Close()
        End If

        If rowscheck = 0 Then
          cn2.Open()
          check_status = "Tamper Open"
          check_date = DateString.ToString
          check_time = TimeString.ToString
          'Check for name and if no existing inmates available, add them
          Dim cmdIns As New MySqlCommand("INSERT INTO alarm (Alrm_TagID,
Alrm_Room,Alrm_TagName,Alrm_Status,alrm_Date,Alrm_Time) VALUES ('" & check_id(j) & "','" & check_room(j) & "','"
& check_name(j) & "','" & check_status & "','" & check_date & "','" & check_time & "')", cn2)
          cmdIns.ExecuteNonQuery()
          If cn2.State = ConnectionState.Open Then
            cn2.Close()
          End If
          'If registration successful, show ID and name
          'Dim style = MsgBoxStyle.Critical

          'MsgBox("!!!!  ALARM !!!!!" & vbCrLf & "Tag ID " & check_id(j) + " Out Of Range Detected",
MsgBoxStyle.Critical, "Alarm")
          alert.lblType.Text = "BABY TAG TAMPER OPEN !!"
          alert.txtID.Text = check_id(j)
          alert.txtName.Text = check_name(j)
          alert.txtLocation.Text = (check_room(j))
          alert.txtTime.Text = check_time
          '  End If ' if found 2
          alert.Show()
          If cn2.State = ConnectionState.Open Then
            cn2.Close()
          End If

        End If

        'End While
        If cnchk.State = ConnectionState.Open Then
          cnchk.Close()
        End If

      End If ' if found

    Next j
    'TmrCall.Enabled = True
    'removetamper()

    'displaylistview()
    'removealarm()
    'clearReader()
    'checkdata2()
    tamper2()

  End Sub
  Private Sub tamper2()

    Dim check_id(1000) As String
```

```
Dim check_id_m(1000) As String
Dim check_name(1000) As String
Dim check_mother(1000) As String
Dim check_room(1000) As String
Dim check_date As String
Dim check_time As String
Dim check_status As String
Dim i As Integer = 0
Dim j As Integer
Dim cn As New MySqlConnection(strCn)
Dim cmd As New MySqlCommand("SELECT * FROM patients", cn)
Dim da As New MySqlDataAdapter(cmd)
Dim ds As New DataSet()
Dim myReader As MySqlDataReader

'Open Drug database
'On Error GoTo errorhandle
If cn.State = ConnectionState.Closed Then
   cn.Open()
End If
'read druglist database and store to buffer
myReader = cmd.ExecuteReader()

While myReader.Read
   i = i + 1
   check_id(i) = myReader.Item("tag_id").ToString
   check_id_m(i) = myReader.Item("tag_m").ToString
   check_name(i) = myReader.Item("name").ToString
   check_mother(i) = myReader.Item("mother").ToString
   check_room(i) = myReader.Item("Room").ToString

End While

If cn.State = ConnectionState.Open Then
   cn.Close()
End If
' complete read druglist

For j = 1 To i

   ' check data in alarm

   Dim cnchk As New MySqlConnection(strCn)
   Dim cmdchk As New MySqlCommand("SELECT * FROM tamper", cnchk)

   Dim dachk As New MySqlDataAdapter(cmdchk)
   Dim dschk As New DataSet()
   Dim myReaderchk As MySqlDataReader
   Dim found As Boolean


   If cnchk.State = ConnectionState.Closed Then
      cnchk.Open()
   End If

   myReaderchk = cmdchk.ExecuteReader()

   While myReaderchk.Read

      'str = myReader.getString(0)

      If check_id_m(j) = myReaderchk.Item("tpr_tagid").ToString Then

         found = True

         Exit While
      Else
```

```
            found = False
        End If

    End While

    If cnchk.State = ConnectionState.Open Then
        cnchk.Close()
    End If



    If found = True Then

        Dim rowscheck As String = 0
        Dim cn2 As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm  WHERE Alrm_TagId = '" & check_id_m(j) &
"'", cn2)
        Dim daCheck As New MySqlDataAdapter(cmdCheck)
        Dim dsCheck As New DataSet()
        Dim dtCheck As New DataTable
        'On Error Resume Next
        If cn2.State = ConnectionState.Closed Then

            cn2.Open()
        End If

        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowscheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        If cn2.State = ConnectionState.Open Then
            cn2.Close()
        End If

        If rowscheck = 0 Then
            cn2.Open()
            check_status = "Tamper Open"
            check_date = DateString.ToString
            check_time = TimeString.ToString
            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO alarm (Alrm_TagID,
Alrm_Room,Alrm_TagName,Alrm_Status,alrm_Date,Alrm_Time) VALUES ('" & check_id_m(j) & "','" & check_room(j) &
"','" & check_name(j) & "','" & check_status & "','" & check_date & "','" & check_time & "')", cn2)
            cmdIns.ExecuteNonQuery()
            If cn2.State = ConnectionState.Open Then
                cn2.Close()
            End If
            'If registration successful, show ID and name
            'Dim style = MsgBoxStyle.Critical

            'MsgBox("!!!!  ALARM !!!!!" & vbCrLf & "Tag ID " & check_id(j) + " Out Of Range Detected",
MsgBoxStyle.Critical, "Alarm")
            alert.lblType.Text = "MOTHER TAG TAMPER OPEN !!"
            alert.txtID.Text = check_id(j)
            alert.txtName.Text = check_name(j)
            alert.txtLocation.Text = (check_room(j))
            alert.txtTime.Text = check_time
```

```
            ' End If ' if found 2
            alert.Show()
            If cn2.State = ConnectionState.Open Then
               cn2.Close()
            End If

          End If

          'End While
          If cnchk.State = ConnectionState.Open Then
             cnchk.Close()
          End If

        End If ' if found

      Next j
      'TmrCall.Enabled = True
      removetamper()
      displaylistview()
      'removealarm()
      'clearReader()
      'checkdata2()


End Sub
Public Sub removealarm()
    Dim cnchk As New MySqlConnection(strCn)
    Dim cmdchk As New MySqlCommand("SELECT * FROM readers", cnchk)

    Dim dachk As New MySqlDataAdapter(cmdchk)
    Dim dschk As New DataSet()
    Dim myReaderchk As MySqlDataReader
    'Dim found As Boolean
    Dim rm_id As String

    If cnchk.State = ConnectionState.Closed Then
       cnchk.Open()
    End If

    myReaderchk = cmdchk.ExecuteReader()

    While myReaderchk.Read


       rm_id = myReaderchk.Item("ReaderID").ToString

       '**********************
       'Check if not in alarm
       Dim cnchk2 As New MySqlConnection(strCn)
       Dim cmdchk2 As New MySqlCommand("SELECT * FROM alarm", cnchk2)

       Dim dachk2 As New MySqlDataAdapter(cmdchk)
       Dim dschk2 As New DataSet()
       Dim myReaderchk2 As MySqlDataReader
       Dim found2 As Boolean
       'Dim rm_id As String

       If cnchk2.State = ConnectionState.Closed Then
          cnchk2.Open()
       End If

       myReaderchk2 = cmdchk2.ExecuteReader()

       While myReaderchk2.Read
```

```
            If rm_id = myReaderchk2.Item("Alrm_TagID").ToString And myReaderchk2.Item("Alrm_Status").ToString = "Out
Range" Then
                found2 = True
                Exit While
            Else
                found2 = False
            End If

        End While
        If cnchk2.State = ConnectionState.Open Then
            cnchk2.Close()
        End If

        '**********************
        If found2 = True Then
            '#####################
            ' Dim id As String = TextBox1.Text.ToString
            Dim rowsCheck As Integer = 0
            Dim cn As New MySqlConnection(strCn)

            Dim cmdCheck As New MySqlCommand("SELECT FROM alarm ", cn)
            Dim daCheck As New MySqlDataAdapter
            Dim dsCheck As New DataSet
            Dim dtCheck As New DataTable


            If rowsCheck = 0 Then
                'Open connection
                cn.Open()

                'Check for name and if no existing inmates available, add them
                Dim cmdIns As New MySqlCommand("DELETE FROM alarm WHERE Alrm_TagID = '" & rm_id & "'", cn)
                rowsCheck = cmdIns.ExecuteReader.RecordsAffected()

                'MsgBox("New ID " & rm_id & " Detected in Range")
                Status_msg.Text = "New ID " & rm_id & " Detected in Range"
                If ConnectionState.Open Then
                    cn.Close()
                End If

                'If registration successful, show ID and name
                'MsgBox(id + "/" + name + " has been registered successfully")


            End If


            '########################
        End If




    End While
    displaylistview()
    displaymatch()
End Sub

Public Sub removetamper()
    Dim cnchk As New MySqlConnection(strCn)
    Dim cmdchk As New MySqlCommand("SELECT * FROM alarm", cnchk)

    Dim dachk As New MySqlDataAdapter(cmdchk)
    Dim dschk As New DataSet()
    Dim myReaderchk As MySqlDataReader
    ' Dim found As Boolean
```

```
Dim rm_id As String
Dim rm_st As String

If cnchk.State = ConnectionState.Closed Then
   cnchk.Open()
End If

myReaderchk = cmdchk.ExecuteReader()

While myReaderchk.Read


   rm_id = myReaderchk.Item("Alrm_TagID").ToString
   rm_st = myReaderchk.Item("Alrm_Status").ToString

   If rm_st = "Tamper Open" Then
      '**********************
      'Check if not in alarm
      Dim cnchk2 As New MySqlConnection(strCn)
      Dim cmdchk2 As New MySqlCommand("SELECT * FROM tamper", cnchk2)

      Dim dachk2 As New MySqlDataAdapter(cmdchk)
      Dim dschk2 As New DataSet()
      Dim myReaderchk2 As MySqlDataReader
      Dim found2 As Boolean
      'Dim rm_id As String
      'On Error Resume Next
      If cnchk2.State = ConnectionState.Closed Then
         cnchk2.Open()
      End If

      myReaderchk2 = cmdchk2.ExecuteReader()

      While myReaderchk2.Read


         If rm_id = myReaderchk2.Item("tpr_Tagid").ToString Then
            found2 = True
            Exit While
         Else
            found2 = False
         End If

      End While
      If cnchk2.State = ConnectionState.Open Then
         cnchk2.Close()
      End If

      '**********************

      If found2 = False Then
         '####################
         ' Dim id As String = TextBox1.Text.ToString
         Dim rowsCheck As Integer = 0
         Dim cn As New MySqlConnection(strCn)

         Dim cmdCheck As New MySqlCommand("SELECT FROM alarm ", cn)
         Dim daCheck As New MySqlDataAdapter
         Dim dsCheck As New DataSet
         Dim dtCheck As New DataTable


         If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            'Check for name and if no existing inmates available, add them
```

```
            Dim cmdIns As New MySqlCommand("DELETE FROM alarm WHERE Alrm_TagID = '" & rm_id & "'", cn)
            rowsCheck = cmdIns.ExecuteReader.RecordsAffected()

            'MsgBox("New ID " & rm_id & " Detected in Range")
            Status_msg.Text = "New ID " & rm_id & " Detected in Range"
            If ConnectionState.Open Then
              cn.Close()
            End If

            'If registration successful, show ID and name
            'MsgBox(id + "/" + name + " has been registered successfully")


          End If


          '########################
        End If
      End If




    End While
    If cnchk.State = ConnectionState.Open Then
      cnchk.Close()
    End If
    displaylistview()

  End Sub
  Private Sub displaylistview()

    Dim i As Integer
    Dim cn As New MySqlConnection(strCn)
    Dim cmd As New MySqlCommand("SELECT * FROM alarm ", cn)
    Dim da As New MySqlDataAdapter(cmd)
    Dim ds As New DataSet()
    Dim myReader As MySqlDataReader

    'Open Drug database

    If cn.State = ConnectionState.Closed Then
      cn.Open()
    End If
    'read druglist database and store to buffer
    ListView1.Items.Clear()

    myReader = cmd.ExecuteReader()

    While myReader.Read
      tagID = ListView1.Items.Add(myReader.Item("Alrm_TagID").ToString)
      tagID.SubItems.Add(myReader.Item("Alrm_TagName").ToString)
      tagID.SubItems.Add(myReader.Item("Alrm_Room").ToString)
      tagID.SubItems.Add(myReader.Item("Alrm_Status").ToString)
      tagID.SubItems.Add(myReader.Item("Alrm_Date").ToString)
      tagID.SubItems.Add(myReader.Item("Alrm_Time").ToString)
    End While



    While i <= ListView1.Items.Count - 1
      If i Mod 2 = 0 Then
        ListView1.Items(i).BackColor = Color.Aquamarine
      Else
        ListView1.Items(i).BackColor = Color.White
```

```
        End If
        i = i + 1
    End While

    If cn.State = ConnectionState.Open Then
        cn.Close()
    End If
    Exit Sub

End Sub
Private Sub displaymatch()

    Dim i As Integer
    'Dim j As Integer
    Dim cn As New MySqlConnection(strCn)
    Dim cmd As New MySqlCommand("SELECT * FROM patients ", cn)
    Dim da As New MySqlDataAdapter(cmd)
    Dim ds As New DataSet()
    Dim myReader As MySqlDataReader
    Dim L1_tag_id As String
    Dim st1 As String = "Not Match"
    Dim st2 As String = "Match"
    Dim found As Boolean = False
    Dim clrchk As String

    'Open Drug database

    If cn.State = ConnectionState.Closed Then
        cn.Open()
    End If
    'read druglist database and store to buffer
    ListView2.Items.Clear()

    myReader = cmd.ExecuteReader()

    While myReader.Read



        lsview = ListView2.Items.Add(myReader.Item("tag_id").ToString)
        lsview.SubItems.Add(myReader.Item("name").ToString)
        lsview.SubItems.Add(myReader.Item("tag_m").ToString)
        lsview.SubItems.Add(myReader.Item("mother").ToString)
        ' lsview.SubItems.Add(myReader.Item(st2).ToString)

        For i = 0 To ListView1.Items.Count - 1
            tagID = ListView1.Items(i)
            L1_tag_id = tagID.SubItems(0).Text


            If L1_tag_id = myReader.Item("tag_id").ToString Or L1_tag_id = myReader.Item("tag_m").ToString Then
                found = True
                Exit For
            Else
                found = False
            End If

        Next i
        If found = True Then
            lsview.SubItems.Add(st1)
        Else
            lsview.SubItems.Add(st2)
        End If

    End While
```

```
    If cn.State = ConnectionState.Open Then
      cn.Close()
    End If


    For i = 0 To ListView2.Items.Count - 1

      tagID = ListView2.Items(i)
      clrchk = tagID.SubItems(4).Text

      If clrchk = st1 Then
        ListView2.Items(i).BackColor = Color.Red
      Else
        ListView2.Items(i).BackColor = Color.Aquamarine
      End If

    Next i

  End Sub

  Private Sub StoreAlarm()
    Dim i As Integer = 0
    Dim idchk As String
    Dim tag_id As String
    Dim aroom As String
    Dim tagname As String
    Dim astatus As String
    Dim adate As String
    Dim atime As String
    Dim found As Boolean = False

    For i = 0 To ListView1.Items.Count - 1
      tagID = ListView1.Items(i)
      tag_id = tagID.SubItems(0).Text
      tagname = tagID.SubItems(1).Text
      aroom = tagID.SubItems(2).Text
      astatus = tagID.SubItems(3).Text
      adate = tagID.SubItems(4).Text
      atime = tagID.SubItems(5).Text

      ' check data in alarm

      Dim cnchk As New MySqlConnection(strCn)
      Dim cmdchk As New MySqlCommand("SELECT * FROM alarm", cnchk)

      Dim dachk As New MySqlDataAdapter(cmdchk)
      Dim dschk As New DataSet()
      Dim myReaderchk As MySqlDataReader



      If cnchk.State = ConnectionState.Closed Then
        cnchk.Open()
      End If

      'Display raw in listview (attendlist)
      myReaderchk = cmdchk.ExecuteReader()

      While myReaderchk.Read

        'str = myReader.getString(0)

        idchk = myReaderchk.Item("Alrm_TagID").ToString
        If tag_id = idchk Then

          found = True
          Exit While
```

```
        Else
            found = False
        End If

    End While

    If cnchk.State = ConnectionState.Open Then
        cnchk.Close()
    End If


    If found = False Then



        Dim rowsCheck As Integer = 0
        Dim ID As String = 0

        Dim cn As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm WHERE Alrm_TagID = '" & tag_id & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()

        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            'MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO alarm (Alrm_TagID,
Alrm_Room,Alrm_TagName,Alrm_Status) VALUES ('" & tag_id & "','" & aroom & "','" & tagname & "','" & astatus & "')", cn)
            cmdIns.ExecuteNonQuery()
            If cn.State = ConnectionState.Open Then
                cn.Close()
            End If

            'If registration successful, show ID and name
            Status_msg.Text = " Alarm .... " + tag_id + " has been detect out of range"

        End If

    End If
```

```
    If cnchk.State = ConnectionState.Open Then
      cnchk.Close()
    End If

    'End While


  Next i

End Sub
Private Sub clearReader()
  ' Dim id As String = TextBox1.Text.ToString
  Dim rowsCheck As Integer = 0
  ' Dim name As String = TextBox2.Text.ToString
  'Dim pswd As String = TextBox3.Text.ToString

  Dim cn As New MySqlConnection(strCn)

  Dim cmdCheck As New MySqlCommand("DELETE FROM readers ", cn)
  Dim daCheck As New MySqlDataAdapter
  Dim dsCheck As New DataSet
  Dim dtCheck As New DataTable

  cn.Open()

  'Check from table (search)
  Try
    With daCheck
      .SelectCommand = cmdCheck
      .Fill(dsCheck, "searchresult")
    End With
    dtCheck = dsCheck.Tables("searchresult")
    rowsCheck = dtCheck.Rows.Count()

    'If unable to connect, show error!
  Catch ex As Exception
    ' MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
  End Try

  'If already connected, close connection
  If ConnectionState.Open Then
    cn.Close()
  End If


  If rowsCheck = 0 Then
    'Open connection
    cn.Open()

    'Check for name and if no existing inmates available, add them
    Dim cmdIns As New MySqlCommand("DELETE FROM readers ", cn)
    rowsCheck = cmdIns.ExecuteReader.RecordsAffected()
    If ConnectionState.Open Then
      cn.Close()
    End If

    'If registration successful, show ID and name
    'MsgBox(id + "/" + name + " has been registered successfully")


  End If


End Sub
```

```
    Private Sub TmrCall_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TmrCall.Tick
        checkdata()
    End Sub


    Private Sub BtnAddOfficer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnAddOfficer.Click
        addofficer.Show()
    End Sub

    Private Sub BtnAddInmate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnAddInmate.Click
        addinm.Show()
    End Sub

    Private Sub BtnLogOut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnLogOut.Click
        officer.Show()
        Me.Hide()
    End Sub



    Private Sub BtnConfig_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnConfig.Click
        frmConfig.Show()
    End Sub

    Private Sub ListView1_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs)
        'MsgBox(" tag " & ListView1.SelectedItems(0).Text)
        'details.TextBox1.Text = ListView1.SelectedItems(0).Text
        'details.Show()
        'MsgBox("listview")
        '

    End Sub

    Private Sub BtnAlert_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnAlert.Click
        AlertHist.Show()
    End Sub

    Private Sub BtnInmLst_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnInmLst.Click
        InmateList.Show()
    End Sub

    Private Sub BtnOpList_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnOpList.Click
        OfficerList.Show()
    End Sub

    Private Sub commStatus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles commStatus.Click

    End Sub

    Private Sub lbl_Operator_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lbl_Operator.Click

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        'MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" + Name + " moved from Zone" + Zone.ToString + " To Zone" +
Location.ToString)
    End Sub
```

```
    Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

    End Sub

    Private Sub PictureBox6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click

    End Sub

    Private Sub ListView1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
        tamper()
    End Sub

    Private Sub TabPage1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub ListView1_MouseDoubleClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles ListView1.MouseDoubleClick
        'MsgBox("List")
        details.TextBox1.Text = ListView1.SelectedItems(0).Text
        details.Show()
    End Sub

    Private Sub ListView1_SelectedIndexChanged_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ListView1.SelectedIndexChanged

        'details.TextBox1.Text = ListView1.SelectedItems(0).Text
        'details.Show()
        'MsgBox("List")
    End Sub

    Private Sub TabControl1_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles
TabControl1.DoubleClick
        'MsgBox("tab control")
    End Sub
End Class
```

Add Baby window in VB

```
Imports MySql.Data.MySqlClient
Public Class addinm

    Private Sub Submit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Submit.Click
        Dim id As String = TextBox1.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim new_room As String = TextBox1.Text.ToString
        Dim new_tagid As String = TextBox2.Text.ToString
        Dim new_babyname As String = TextBox3.Text.ToString
        Dim new_tagmom As String = TextBox4.Text.ToString
        Dim new_momname As String = TextBox5.Text.ToString
        Dim new_date As String = TextBox6.Text.ToString
        Dim new_time As String = TextBox7.Text.ToString
        Dim strCn As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="

        'Connect to database
        Dim cn As New MySqlConnection(strCn)

        'Open database
        Dim cmdCheck As New MySqlCommand("SELECT * FROM patients WHERE tag_id = '" & new_tagid & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable
        cn.Open()

        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If
        If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            'Check for name and if no existing patientss available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO patients (name, room, mother, tag_m, tag_id, P_Date, P_Time)
VALUES ('" & new_babyname & "','" & new_room & "','" & new_momname & "','" & new_tagid & "','" & new_tagmom & "','"
& new_date & "','" & new_time & "')", cn)
            cmdIns.ExecuteNonQuery()

            If ConnectionState.Open Then
                cn.Close()
            End If
            'If registration successful, show ID and name
            MsgBox(new_tagid + "/" + new_babyname + " has been registered successfully")
        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + new_tagid + "is already registered in the database")

        End If
    End Sub
```

Add User window in VB

```
mports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class addofficer

    Dim strCn As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="

    Private Sub addofficer_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load


    End Sub


    Private Sub btn_confirm_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_confirm.Click
        Dim id As String = TextBox1.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim name As String = TextBox2.Text.ToString
        Dim pswd As String = TextBox3.Text.ToString

        Dim cn As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("SELECT * FROM officer WHERE id = '" & id & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()

        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            'Check for name and if no existing patientss available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO officer (id, name, Password) VALUES ('" & id & "','" & name
& "','" & pswd & "')", cn)
            cmdIns.ExecuteNonQuery()
            If ConnectionState.Open Then
                cn.Close()
            End If
```

```
        'If registration successful, show ID and name
        MsgBox(id + "/" + name + " has been registered successfully")

      Else
        'If already existing officer, show error
        MsgBox("Error!!! " + id + "is already registered in the database")

      End If

    End Sub


    Private Sub btn_Oclose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_Oclose.Click
      Me.Close()
    End Sub

    Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub
End Class
```

Babies list in VB

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class InmateList

    Dim strCn As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub InmateList_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        InmList.Columns.Clear()
        InmList.Columns.Add("No", 30, HorizontalAlignment.Left)
        InmList.Columns.Add("Room", 60, HorizontalAlignment.Left)
        InmList.Columns.Add("Baby Tag", 60, HorizontalAlignment.Left)
        InmList.Columns.Add("Baby Name", 180, HorizontalAlignment.Left)
        InmList.Columns.Add("Mother Tag", 60, HorizontalAlignment.Left)
        InmList.Columns.Add("Mother Name", 180, HorizontalAlignment.Left)
        InmList.Columns.Add("Date", 70, HorizontalAlignment.Left)
        InmList.Columns.Add("Time", 60, HorizontalAlignment.Left)

        ' Strt = New Thread(AddressOf Thread1)
        'Strt.Start()
        Thread1()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM patients", cn)
            Dim myReader As MySqlDataReader


            InmList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            myReader = cmd.ExecuteReader()
            While myReader.Read
                No = No + 1
                ID = InmList.Items.Add(No.ToString)
                ID.SubItems.Add(myReader.Item("Room").ToString)
                ID.SubItems.Add(myReader.Item("Tag_Id").ToString)
                ID.SubItems.Add(myReader.Item("name").ToString)
                ID.SubItems.Add(myReader.Item("tag_m").ToString)
                ID.SubItems.Add(myReader.Item("mother").ToString)
                ID.SubItems.Add(myReader.Item("P_Date").ToString)
                ID.SubItems.Add(myReader.Item("P_Time").ToString)


            End While

            cmd.Connection.Close()

            'coloring backgroung
            While i <= InmList.Items.Count - 1
                If i Mod 2 = 0 Then
```

```
            InmList.Items(i).BackColor = Color.Aquamarine
          Else
            InmList.Items(i).BackColor = Color.White
          End If
          i = i + 1
        End While

      End If

    End Sub

    Sub Thread1()
      list()
    End Sub

    Private Sub InmList_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
InmList.SelectedIndexChanged

    End Sub
End Class
```

Personnel list in VB

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class OfficerList

    Dim strCn As String = "Database=baby_db;Data Source=localhost;User Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub OfficerList_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        OffList.Columns.Clear()
        OffList.Columns.Add("No", 90, HorizontalAlignment.Left)
        OffList.Columns.Add("ID", 90, HorizontalAlignment.Left)
        OffList.Columns.Add("Name", 180, HorizontalAlignment.Left)
        'Strt = New Thread(AddressOf Thread1)
        'Strt.Start()
        Thread1()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM officer", cn)
            Dim myReader As MySqlDataReader


            OffList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If
            myReader = cmd.ExecuteReader()
            While myReader.Read
                No = No + 1
                ID = OffList.Items.Add(No.ToString)
                ID.SubItems.Add(myReader.Item("id").ToString)
                ID.SubItems.Add(myReader.Item("Name").ToString)
            End While
            cmd.Connection.Close()
            'coloring backgroung
            While i <= OffList.Items.Count - 1
                If i Mod 2 = 0 Then
                    OffList.Items(i).BackColor = Color.Aquamarine
                Else
                    OffList.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While
        End If
    End Sub
    Sub Thread1()
        list()
    End Sub
    Private Sub OffList_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
OffList.SelectedIndexChanged

    End Sub
End Class
```

Tag out of zone in VB

```
Public Class alert

    Private Sub alert_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'Dim UserQueryResult As DataTable = officer.DoSQL("SELECT * FROM `patients` WHERE `tag_id` = '" & Me.Tag & "' LIMIT 0,1")

        'Dim buffer(3) As String
        'buffer(1) = UserQueryResult.Rows(0)(1)
        'buffer(2) = UserQueryResult.Rows(0)(0)
        'buffer(3) = UserQueryResult.Rows(0)(2)

        'lbl_nm.Text = buffer(1)
        'lbl_d.Text = buffer(2)
        'lbl_loca.Text = buffer(3)
        'lbl_tm.Text = Now.ToShortTimeString
    End Sub

    Private Sub btn_Submit_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_Submit.Click
        Me.Close()
    End Sub

    Private Sub txtID_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub LogoPictureBox_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

        If lblType.Enabled = True Then
            lblType.Enabled = False
        Else
            lblType.Enabled = True
        End If

    End Sub
End Class
```

APPENDIX B

GANTT CHART FOR FYP1

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Selection of Project Topic	■	■						Mid-semester break							
2	Preliminary Research Work		■	■	■											
3	Submission of Preliminary Report				●											
4	Seminar 1 (optional)					■	■	■								
5	Project Work					■	■	■								
6	Submission of Progress Report									●						
7	Seminar 2 (compulsory)										■	■	■	■		
8	Project work continues									■	■	■	■	■		
9	Submission of Interim Report Final Draft														●	
10	Oral Presentation															●

● Suggested milestone

■ Process

GANTT CHART FOR FYP2

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Project Work Continue	■	■	■					Mid-Semester Break							
2	Submission of Progress Report 1				●											
3	Project Work Continue				■	■	■	■								
4	Submission of Progress Report 2									●						
5	Seminar (compulsory)										■	■	■			
5	Project work continue									■	■	■	■			
6	Poster Exhibition											●				
7	Submission of Dissertation (soft bound)													●		
8	Oral Presentation														●	
9	Submission of Project Dissertation (Hard Bound)															●

● Suggested milestone

■ Process